# Mục lục

[Mục lục 1](#_Toc443978248)

[LỜI CẢM ƠN 2](#_Toc443978249)

[MỞ ĐẦU 3](#_Toc443978250)

[CHƯƠNG 1. TÌM HIỂU PHP & MYSQL 5](#_Toc443978251)

[1.1. Giới thiệu PHP 5](#_Toc443978252)

[1.2. Cài đặt PHP, Apache, MySQL 5](#_Toc443978253)

[CHƯƠNG 2. TÌM HIỂU WORDPRESS 9](#_Toc443978254)

[2.1. Giới thiệu WordPress 9](#_Toc443978255)

[2.2. Những lý do nên sử dụng WordPress 9](#_Toc443978256)

[2.3. Giới thiệu WordPress.org 11](#_Toc443978257)

[2.4. Giới thiệu WordPress.com 12](#_Toc443978258)

[2.5. Cài đặt WordPress trên Localhostss 13](#_Toc443978259)

[2.6. Tìm hiểu cách vận hành của WordPress 20](#_Toc443978260)

[CHƯƠNG 3. TÌM HIỂU CÁCH XÂY DỰNG PLUGIN TRONG WORDPRESS 24](#_Toc443978261)

[3.1. Tổng quan 24](#_Toc443978262)

[3.2. Bắt đầu viết plugin 25](#_Toc443978263)

[3.3. Tìm hiểu về Hook 28](#_Toc443978264)

[3.4. Lập trình hướng đối tượng và cách đặt tên hàm 28](#_Toc443978265)

[CHƯƠNG 4. XÂY DỰNG PLUGIN DEMO 30](#_Toc443978266)

# LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy cô giáo, gia đình và bạn bè đã động viên, giúp đỡ em rất nhiều trong quá trình hoàn thành bản báo cáo này. Đặc biệt em xin bày tỏ lòng cảm ơn sâu sắc tới thầy giáo hướng dẫn Thạc sỹ Nguyễn Tuấn Anh về sự chỉ dẫn tận tình và tận tâm hướng dẫn em từ những ý tưởng ban đầu tới khi hoàn thành báo cáo này.

Em rất mong đón nhận sự đánh giá, bổ sung và những lời chỉ bảo của các thầy cô giúp em có thể tiếp tục nghiên cứu kĩ hơn về lĩnh vực này.

Em xin chân thành cảm ơn!

Thái Nguyên, tháng 1/2016

Sinh Viên

**Trương Tuyến Anh**

# MỞ ĐẦU

**1.Lý do chọn đề tài**

Với sự phát triển nhảy vọt của công nghệ thông tin hiện nay, Internet ngày càng có vai trò quan trọng trong lĩnh vực khoa học, kĩ thuật và đời sống. Với mạng Internet, ngành tin học đã thực sự tạo nên cuộc cách mạng trao đổi thông tin trong mọi lĩnh vực văn hóa, xã hội, chính trị, kinh tế …

Ngày nay, cách đơn giản nhất để tiếp cận Internet chính là thông qua các trang web. Hiện nay việc làm một website để quảng bá công ty, để bán hàng, hay chỉ là một blog cá nhân … trở nên rất dễ dàng và đơn giản mà không cần nhiều kiến thức chuyên sâu.

Website sẽ giúp bạn đưa những thông tin mong muốn của mình lên mạng Internet cho mọi người cùng xem một cách dễ dàng. Sự ra đời của các ngôn ngữ lập trình cho phép chúng ta xây dựng trang web có thể đáp ứng được các yêu cầu từ người sử dụng. PHP(Personal Home Page) là kịch bản phía trình chủ (Server Script) cho phép chúng ta xây dựng trang web trên hệ quản trị cơ sở dữ liệu MySQL. Đặc biệt với sự ra đời của WordPress, một CMS (Content Manage System) mạnh mẽ viết trên ngôn ngữ PHP đã giúp cho việc sở hữu một trang web đáp ứng mọi yêu cầu của người dùng trở nên dễ dàng hơn bao giờ hết.

Vì lý do đó, được sự hướng dẫn và giúp đỡ của thầy giáo Thạc sĩ Nguyễn Tuấn Anh, em đã chọn đề tài: "Tìm hiều WordPress và xây dựng plugin demo" làm đề tài cho đợt thực tập cơ sở ngành này.

**2.Mục tiêu nghiên cứu**

Nghiên cứu cách sử dụng WordPress và xây dựng plugin trong WordPress

**3.Bố cục bản báo cáo**

**Chương I:** Tìm hiểu PHP & MySQL và WordPress

**Chương II:** Tìm hiểu cách xây dựng plugin trong WordPress

**Chương III:** Xây dựng một plugin demo đơn giản cho WordPress

# TÌM HIỂU PHP & MYSQL VÀ WORDPRESS

## Giới thiệu PHP

PHP (Hypertext Preprocessor) là ngôn ngữ script trên server được thiết kế để dễ dang xây dựng các trang web động. Mã PHP có thể thực thi trên webserver để tạo ra mã HTMl và xuất ra trình duyệt web theo yêu cầu người sử dụng

Ngôn ngữ PHP ra đời năm 1994 bởi Rasmus Lerdorf sau đó được phát triển bởi nhiều người, trải qua nhiều phiên bản. Phiên bản PHP 5 được công bố 7/2004 hiện nay bản PHP 7 đang được thử nghiệm.

Có rất nhiều lý do khiến việc sử dụng ngôn ngữ này chiếm ưu thế, sau đây là một số lý do cơ bản:

-Mã nguồn mở (Open Source Code)

-Miễn phí, download và cài đặt dễ dàng

-Ngôn ngữ dễ học, dễ viết

-Mã nguồn không phải sửa nhiều khi chạy trên các hệ điều hành: Windows, Linux, Unix…

-Rất đơn giản trong việc kết nối với nhiều hệ quản trị cơ sở dữ liệu như: MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL. Adabas, dBase, Empress, FilePro Infomix, InterBase, mSQL, Solid, Sybase, Velocis và nhiều hệ thống cơ sở dữ liệu khác

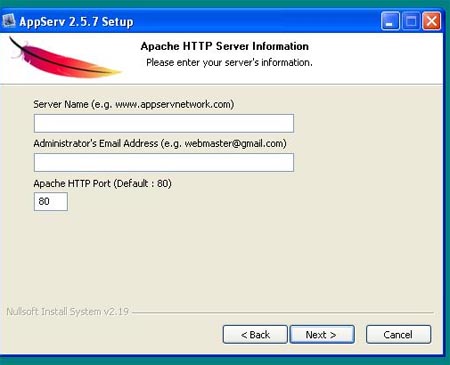
## Cài đặt PHP, Apache, MySQL

Để cài đặt PHP, chúng ta có thể cài đặt từng gói riêng lẻ (PHP, Apache, Mysql). Tuy nhiên, cài đặt dạng gói tích hợp sẽ tiện lợi cho việc sau này hơn rất nhiều.

Ở đây chúng ta sử dụng gói phần mềm: Appserv - Win - 2.5.8 (Tên phần mềm là Appserv, dành cho phiên bản window, phiên bản 2.5.8).

**Bước 1:** C đặt bình thường bằng cách nhấp vào file exe.

**Bước 2:** Phần mềm sẽ cho chúng ta chọn cần cài những module nào. Hãy giữ nguyên tất cả như hình dưới và nhấn next.



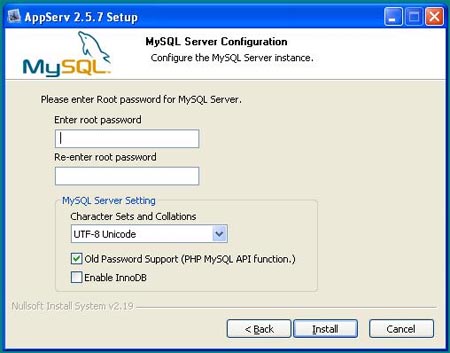
**Bước 3:** Trong giao diện dưới:

-Server Name: localhost

-Email: nhập vào một email:

-Port: Mặc định là 80, tuy nhiên nếu máy đã cài IIS thì có thể chỉnh thành 81 để chạy dịch vụ bên appserv.

-Nhấn next để qua trang kế tiếp.



**Bước 4:** Trong giao diện bên dưới ta điền thông tin như sau:

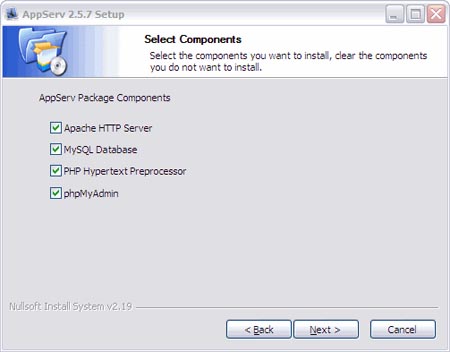
-Enter root password: gõ vào root

-Re-enter root password: nhập lại 1 lần nữa root

Các cấu hình bên dưỡi giữ nguyên.

Ở phần: Enable InnoDB đánh dấu vào. Để MYSQL sử lý được các ứng dụng có bật chế độ InnoDB.

Nhấn next để hoàn tất việc cài đặt.

  
  
Sau khi cài đặt xong gõ vào trình duyệt : **http://localhost**mà ra giao diện bên dưới, tức là chúng tass đã cài đặt thành công **appserv**.



Như vậy là chúng ta đã cài đặt PHP thành công.

## Giới thiệu WordPress

WordPress là một phần mềm nguồn mở (Open Source Software) được viết bằng ngôn ngữ lập trình website PHP (Hypertext Preprocessor) và sử dụng hệ quản trị cơ sở dữ liệu MySQL. WordPress được ra mắt lần đầu tiên vào ngày 27/5/2003 bởi tác giả Matt Mullenweg và Mike Little.

WordPress là một mã nguồn mở bằng ngôn ngữ PHP để hỗ trợ tạo blog cá nhân, và nó được rất nhiều người sử dụng ủng hộ về tính dễ sử dụng, nhiều tính năng hữu ích. Qua thời gian, số lượng người sử dụng tăng lên, các cộng tác viên là những lập trình viên cũng tham gia đông đảo để phát triển mã nguồn WordPress có thêm những tính năng tuyệt vời.

Hiện nay, WordPress đã được xem như là một hệ quản trị nội dung (CMS – Content Management System) vượt trội để hỗ trợ người dùng tạo ra nhiều thể loại website khác nhau như blog, website tin tức/tạp chí, giới thiệu doanh nghiệp, bán hàng – thương mại điện tử, thậm chí với các loại website có độ phức tạp cao như đặt phòng khách sạn, thuê xe, đăng dự án bất động sản,…

Hầu như mọi hình thức website với quy mô nhỏ và vừa đều có thể triển khai trên nền tảng WordPress. Nhưng như thế không có nghĩa là WordPress chỉ thích hợp với các dự án nhỏ, mà hiện nay có tới khoảng 25% website trong danh sách 100 website lớn nhất thế giới sử dụng mã nguồn WordPress. Ví dụ như trang tạp chí TechCrunch, Mashable, CNN, BBC America, Variety, Sony Music, MTV News, Bata, Quartz,…

WordPress là một bộ mã nguồn mở, có lẽ vì điều này mà tạo nên sự thành công của WordPress như ngày hôm nay. Điều này có nghĩa là WordPress được tạo ra bởi cộng đồng và phục vụ cho lợi ích của cộng đồng, nó được phát triển bởi hàng trăm tình nguyện viên trên thế giới và rất rất nhiều lập trình viên khác đã đóng góp vào đây. Và để sử dụng WordPress thì không cần phải trả bất kỳ chi phí nào cho bộ mã nguồn này.

## Những lý do nên sử dụng WordPress

+ Dễ sử dụng

WordPress được phát triển nhằm phục vụ đối tượng người dùng phổ thông, không có nhiều kiến thức về lập trình website nâng cao. Các thao tác trong WordPress rất đơn giản, giao diện quản trị trực quan giúp người dùng có thể nắm rõ cơ cấu quản lý một website WordPress trong thời gian ngắn. Về cách cài đặt lại càng dễ hơn, người dùng có thể tự cài đặt một website WordPress trên host (máy chủ) riêng của mình và tự vận hành nó sau vài cú click.

+ Cộng đồng hỗ trợ đông đảo

Là một mã nguồn CMS mở phổ biến nhất thế giới, điều này cũng có nghĩa là người dùng sẽ được cộng đồng người sử dụng WordPress hỗ trợ các khó khăn gặp phải trong quá trình sử dụng. Nếu có khả năng tiếng Anh tốt, họ có thể dễ dàng tìm câu trả lời cho vấn đề đang gặp phải trên Google chỉ với vài từ khóa tìm kiếm.

+ Nhiều gói giao diện có sẵn

Trong khi sử dụng WordPress, khái niệm giao diện cho website WordPress thường được gọi là theme. Hiện nay WordPress có rất nhiều theme miễn phí khác nhau cho website tùy theo nhu cầu sử dụng.

+ Nhiều plugin hỗ trợ

Plugin nghĩa là một trình cắm thêm vào website để bổ sung các chức năng cần thiết. Hiện tại WordPress có hàng triệu plugin trên trang chủ, đáp ứng hầu hết nhu cầu của người sử dụng. Người dùng còn có thể dễ dàng phát triền các plugin mới, hoặc plugin đã có nếu có kiến thức PHP

+ Dễ phát triển cho lập trình viên

Với hàng nghìn hàm (function) có sẵn, các lập trình viên có thể thoải mái sử dụng, họ cũng có thể thay đổi cấu trúc của một hàm, tương tác với các hook mà không cần sửa đổi trực tiếp trong mã nguồn của WordPress và hầu như quy trình làm việc của một lập trình viên chuyên nghiệp có thể ứng dụng dễ dàng vào WordPress.

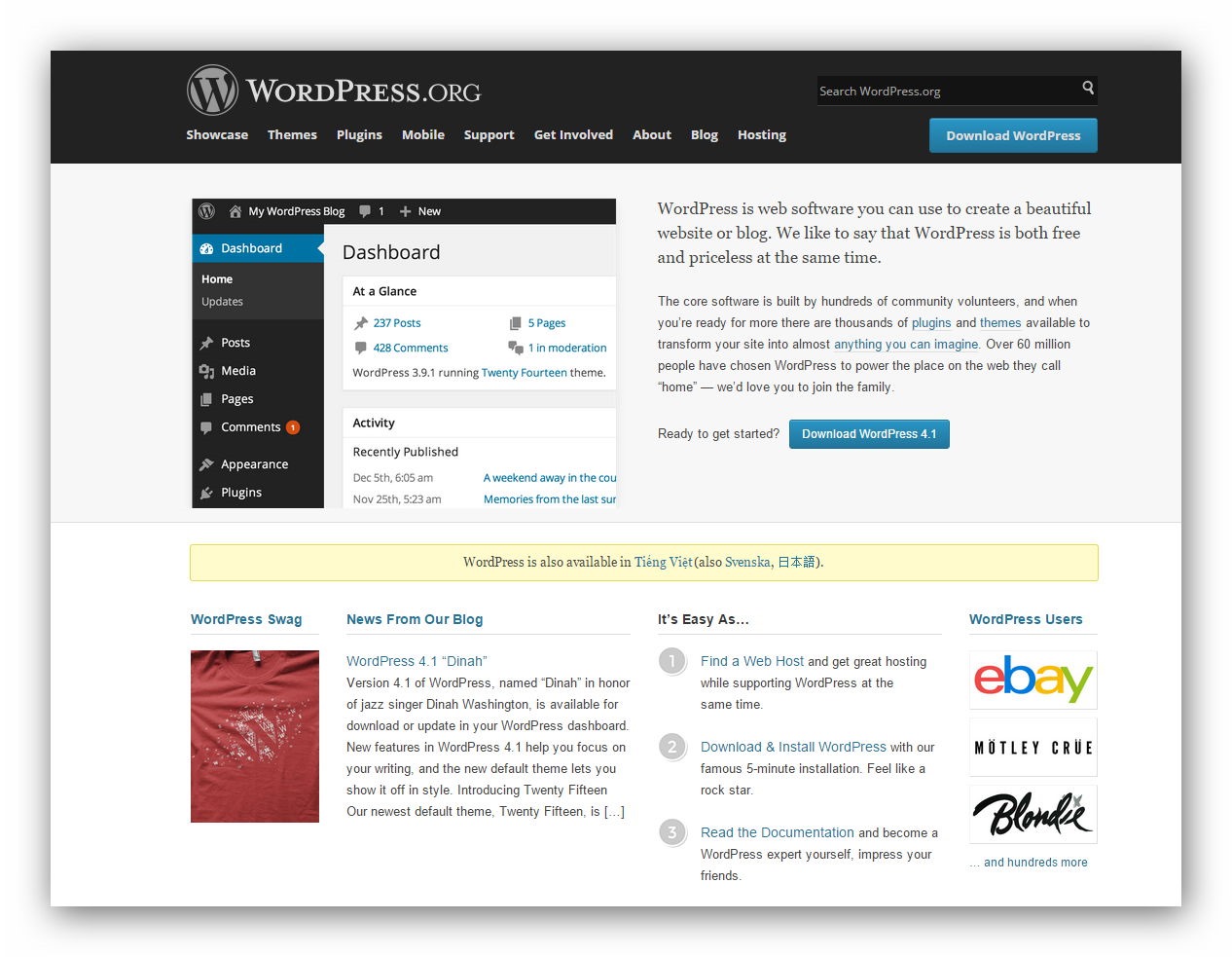
+ Hỗ trợ nhiều ngôn ngữ

Mã nguồn WordPress hiện tại có rất nhiều gói ngôn ngữ đi kèm, bao gồm tiếng Việt. Mặc dù trong mỗi giao diện hay plugin đều có ngôn ngữ riêng nhưng chúng ta có thể dễ dàng tự dịch lại nó với các phần mềm hỗ trợ.

+ Có thể làm nhiều loại website

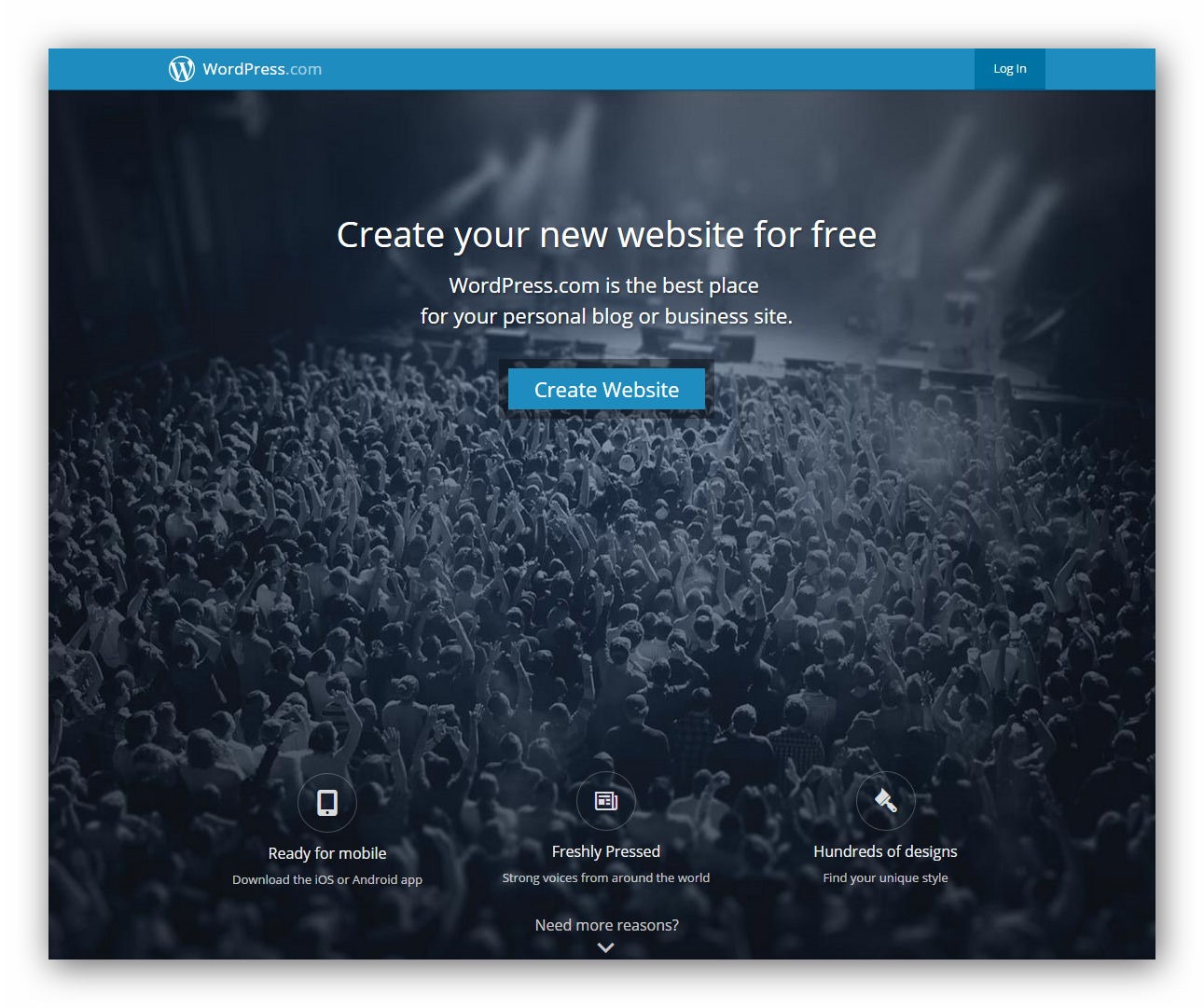
Dùng WordPresschúng takhông chỉ có thể làm blog cá nhân, mà còn có thể biến website thành một trang bán hàng, một website giới thiệu công ty, một tờ tạp chí online bằng việc sử dụng kết hợp các theme và plugin với nhau.

## Giới thiệu WordPress.org



WordPress.org là một trong những nơi góp phần tạo ra những blogger nổi tiếng trên khắp thế giới hiện nay. Khi đến với dịch vụ này, chúng ta có thể tải bộ cài đặt của WordPress cũng như tải các plugin và giao diện do cộng đồng cung cấp. Để có thể sử dụng được dịch vụ này thì chúng ta cần phải có một tên miền và một hosting, và hosting của phải đáp ứng được cấu hình tối thiểu để cài đặt WordPress.

## Giới thiệu WordPress.com



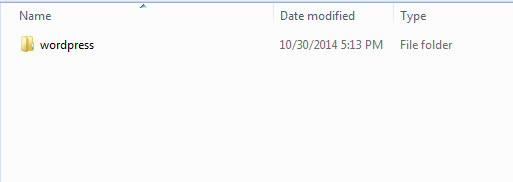
Có thêm một dịch vụ nữa của WordPress và được gọi với tên WordPress.com cho phép người dùng tạo một blog với bản chất là hoàn toàn miễn phí bởi không cần bỏ ra bất cứ đồng nào để mua tên miền cũng như hosting. Khi đăng ký blog miễn phí trên dịch vụ này, người dùng sẽ được sử dụng tên miền con của WordPress và những tiện ích cơ bản của blog WordPress. Nhưng khả năng tùy chỉnh thì không được như WordPress self hosted.

## Cài đặt WordPress trên Localhostss

**Bước 1. Tải mã nguồn từ website WordPress.org**

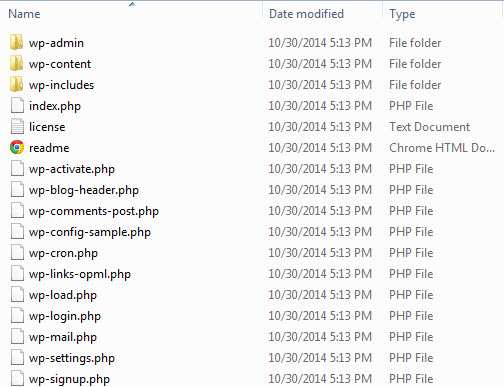
Trước tiên chúng ta hãy tải phiên bản mới nhất của mã nguồn WordPress tại địa chỉ  https://wordpress.org/latest.zip.

Sau đó giải nén ra sẽ có được một thư mục mang tên “wordpress“. Có thể thư mục wordpress này sẽ được lồng trong một thư mục khác tên là wordpress-x (x ở đây là số phiên bản), nhưng nói chung chúng ta sẽ có được một thư mục tên wordpress như hình dưới.

[](http://s3cdn.thach.io/wp-content/uploads/2013/09/18233818/cai-dat-wordpress-localhost-01.jpg)

Thư mục sau khi giải nén mã nguồn

Tiếp tục, hãy truy cập vào thư mục wordpress, chúng ta sẽ thấy có một số thư mục tên là wp-admin, wp-includes, wp-content và một số tập tin tên là index.php, wp-config-sample.php,…Tất cả tập tin và thư mục này, chúng ta gọi nó là mã nguồn WordPress.

[](http://s3cdn.thach.io/wp-content/uploads/2013/09/18233819/cai-dat-wordpress-localhost-02.jpg)

**Bước 2. Copy mã nguồn WordPress vào Localhost**

Bây giờ, hãy copy toàn bộ file và thư mục này vào thư mục website của bạn trong localhost (ví dụ:**C:\Appserv\www\wordpress**). Khi này website của chúng ta có địa chỉ **http://localhost /wordpress/.**

**Bước 3. Tạo mới một database**

Để chạy được WordPress thì localhost của bạn phải có một database dùng MySQL để nó có thể lưu các dữ liệu mềm vào đó như bài viết, các thiết lập,…trên website. Chúng ta truy cập vào phpMyadmin để tạo mới một cơ sở dữ liệu

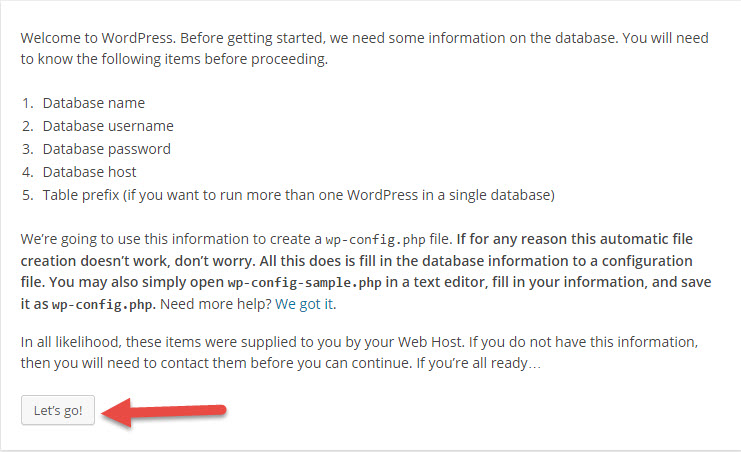
**Bước 4. Chạy website để cài đặt**

Truy cập vào website với đường dẫn **http://localhost/wordpress**

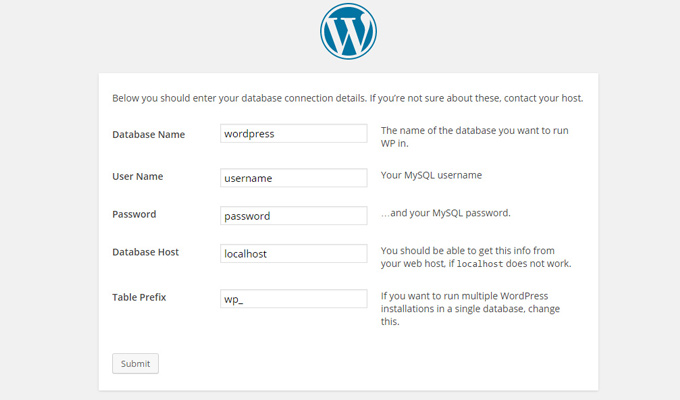
Lúc này, nó sẽ hiện ra bảng chọn ngôn ngữ cần cài đặt cho WordPress, hãy chọn  English hoặc tiếng việt và ấn Continue.



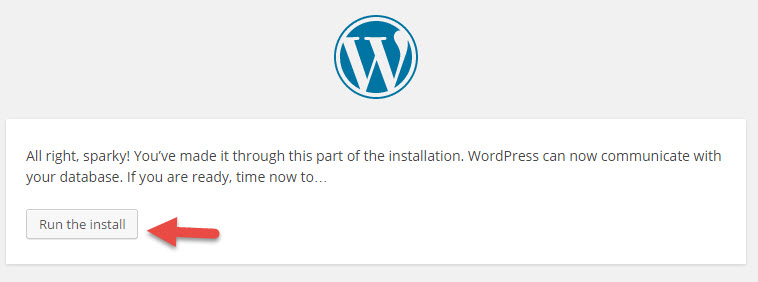
Ở bước tiếp theo, nó sẽ nhắc nhở cho bạn là chưa tiến hành đổi file **wp-config-sample.php** thành **wp-config.php** và khai báo thông tin database vào đó. Hãy ấn  **Let’s Go** để nó tự làm việc đó cho bạn.



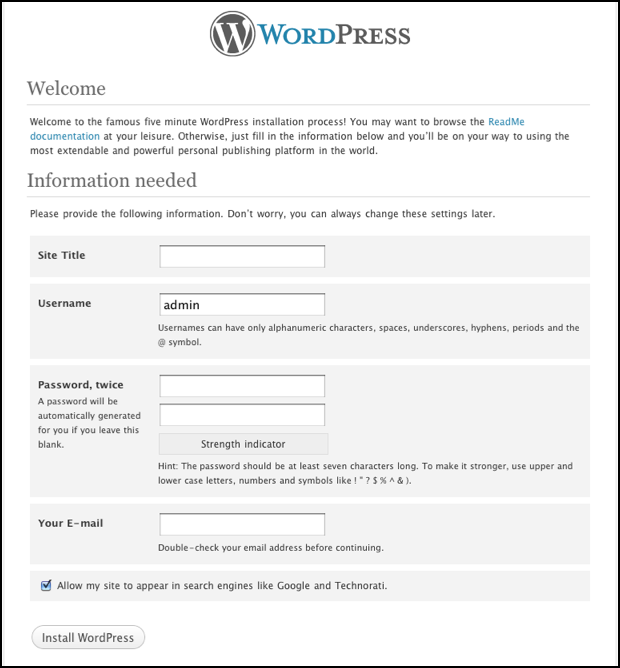
Tiếp theo ta tiến hành nhập thông tin Database



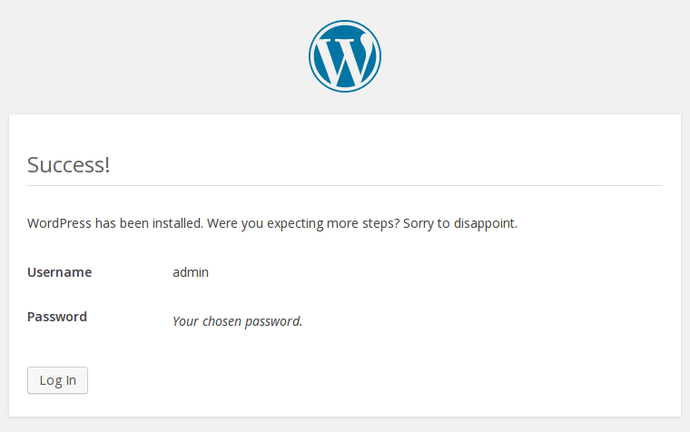
Khi nhập xong thông tin database, hãy ấn nút **Submit** để làm bước kế tiếp. Nếu bước kế tiếp nó hiện ra như hình dưới thì nghĩa là bạn đã nhập thông tin database chính xác, hãy ấn nút **Run the install** để bắt đầu cài đặt.



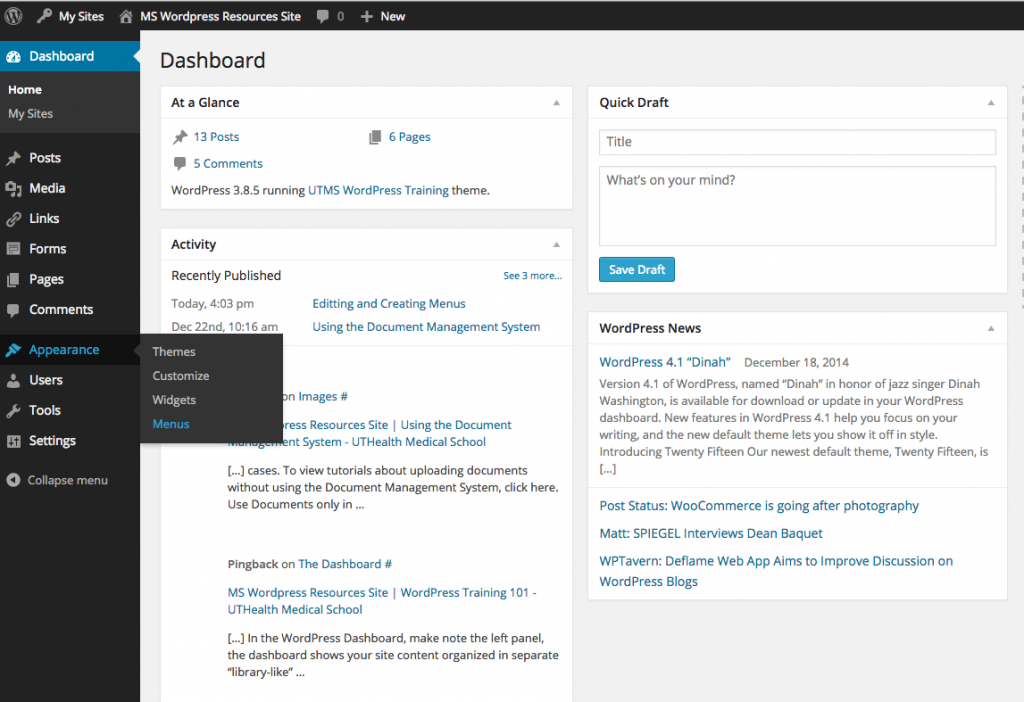
Ở bước cài đặt này, các bạn sẽ cần phải thiết lập các thông tin quan trọng cho website như Tên của website, tên tài khoản admin cùng mật khẩu,…Nhập xong hãy ấn **nút Install WordPress.**



Và nếu nó hiện chữ Success! như thế này là bạn đã cài đặt thành công, click vào nút Log in để đăng nhập vào bảng quản trị WordPress.



Và đây là giao diện trang quản trị khi đăng nhập vào WordPress

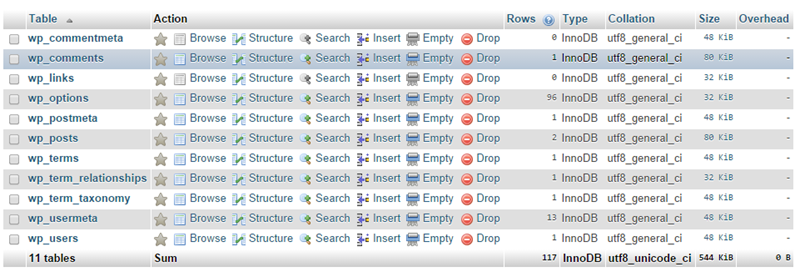


## Tìm hiểu cách vận hành của WordPress

**\* Các dữ liệu làm việc thế nào?**

Trong mã nguồn WordPress, nó đã được lập trình nhiều tính năng giúp bản thân nó có thể tương tác với cơ sở dữ liệu (database) như MySQL để giúp người sử dụng có thể lưu trữ dữ liệu mềm trên website. Tất cả các dữ liệu mềm được lưu trữ vào database sẽ bao gồm các nội dung văn bản được nhập vào website, các thiết lập (vì các thiết lập sẽ lưu dưới dạng một kiểu dữ liệu) và một số dữ liệu khác.

Nếu xem database thông qua phpMyAdmin hoặc các ứng dụng tương tự, chúng ta sẽ thấy WordPress có nhiều bảng dữ liệu để chứa các dữ liệu được lưu vào.



Rồi sau đó, các tính năng trong mã nguồn sẽ tương tác với các bảng dữ liệu đó và đưa ra ngoài trình duyệt hiển thị dưới dạng HTML thông qua các tập tin đã được thiết lập khuôn mẫu trong theme (ta gọi là template).

**\*WordPress làm gì mỗi khi có người truy cập vào website?**

Khi một người truy cập vào website, WordPress sẽ tự động xử lý theo quy trình sau:

*Bước 1 – Khởi động mã nguồn*

Tập tin **index.php**trong mã nguồn (không phải của theme hay plugin) được tải ra, sau đó nó sẽ truy xuất các tập tin cốt lõi khác như **wp-config.php** vốn để kết nối đến cơ sở dữ liệu, **wp-settings.php**,…. Khi mở tập tin **index.php** lên xem và lần mò theo các tập tin được nhúng vào sẽ biết được quy trình nó load theo thứ tự các tập tin.

Ở bước này, WordPress sẽ kết nối đến cơ sở dữ liệu được thiết lập trong **wp-config.php**, sau đó sẽ tiến hành tải những tính năng trong mã nguồn như **/wp-include/functions.php, /wp-include/options.php,**…nhằm nạp sẵn các chức năng cần thiết để có thể tiếp nhận những dữ liệu.

Và cũng trong bước này, toàn bộ các dữ liệu được lưu vào bảng **wp\_options** có giá trị ở cột **autoload** là **true** sẽ được truy xuất ra hết, dù thiết lập đó có được sử dụng hay không. Mục đích của bước này là mang sẵn các tuỳ chọn của website cùng các plugin để hỗ trợ cho các bước sau, đặc biệt là bước tải plugin để nó làm việc chính xác.

*Bước 2 – Kích hoạt plugin*

Các plugin đang kích hoạt trong website sẽ được tải ra ngay sau khi mã nguồn WordPress khởi động xong. Bởi vì các tính năng trong WordPress Core được sử dụng trong plugin thường sẽ được gắn vào hook **init** (là một điểm neo để kích hoạt kịch bản nào đó) nên nó sẽ load ngay sau khi WordPress khởi động.

*Bước 3 – Thực thi tập tin functions.php trong theme*

Lúc này, WordPress sẽ tiến hành dò tìm đến tập tin **functions.php** trong theme đang được kích hoạt để tải các tính năng mà người tạo ra theme đã khai báo ở đó. Vậy làm sao WordPress có thể hiểu được website đang dùng theme nào? Đó là ở bước 1, nó đã kết nối vào database và dựa theo khoá **current**\_theme trong cột **option\_name** tại bảng **wp\_options**.

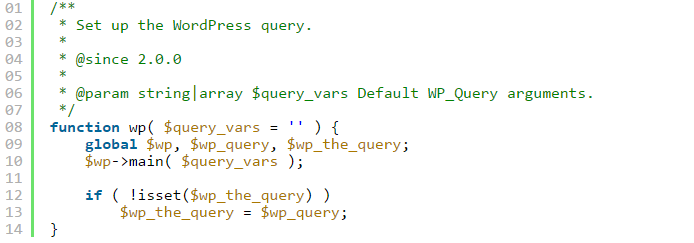


*Bước 4 – Phân tích truy vấn và khởi tạo truy vấn*

Đây là bước quan trọng để website có thể hiển thị nội dung ra bên ngoài, vì các nội dung sẽ được trả về sau khi các truy vấn gửi vào database.

Truy vấn nghĩa là một mệnh lệnh được gửi vào database nhằm lấy các thông tin mà truy vấn đó cần xem. Truy vấn ở đây là truy vấn SQL gửi vào MySQL Server.

Trước tiên, WordPress sẽ chạy hàm **wp()**được thiết lập **trong /wp-include/functions.php** vốn để gọi phương thức **$wp->main()** cho mục đích thiết lập truy vấn. Đối tượng **$wp** được tạo ra bởi lớp **WP** trong **/wp-include/class-wp.php.**



Lúc này, phương thức **$wp->parse\_request()** sẽ được khởi động cho mục đích phân tích truy vấn dựa theo đường dẫn của website. Vì WordPress sẽ tự động sinh truy vấn dựa theo đường dẫn, ví dụ khi chúng ta vào **http://domain.com/?p=123** thì WordPress sẽ gửi một truy vấn vào database để lấy dữ liệu của **post** mang số **ID** là **123**.

Sau khi truy vấn được phân tích, WordPress sẽ làm việc tiếp theo là thiết lập các hàm điều kiện thông qua phương thức **$wp\_query->parse\_query().** Sau đó nó sẽ chuyển các truy vấn đã được tạo ra thành truy vấn bằng các lệnh SQL nhằm gửi đến MySQL để lấy dữ liệu bài viết bằng phương thức **$wp\_query->get\_posts().** Nếu database có dữ liệu, các bài viết sẽ được lấy về sau khi gửi truy vấn và nó sẽ được lưu vào đối tượng**$wp\_query** để nó sử dụng trong các vòng lặp cho việc hiển thị bài viết.

Trong quá trình gửi truy vấn này, nếu nó không tìm thấy dữ liệu thì sẽ phân tích và hiển thị báo lỗi **404**.

Và cuối cùng là nó có dữ liệu, thì nó sẽ thiết lập biến **$post**để sử dụng trong vòng lặp. Biến **$post** là đối tượng chứa các dữ liệu của bài viết thông qua các thuộc tính.

*Bước 5 – Thực thi các tập tin khuôn mẫu (template) trong theme*

Sau khi **WordPress** đã có dữ liệu bài viết và các dữ liệu liên quan mà nó đã làm ở bước 4, thì nó sẽ tiến hành xử lý các tập tin template của theme thông qua cấu trúc template. Sau đó các nội dung và trang chủ của website sẽ được hiển thị dựa theo các template tags dưới dạng HTML.

# TÌM HIỂU CÁCH XÂY DỰNG PLUGIN TRONG WORDPRESS

## Tổng quan

\* Plugin là gì

Trong WordPress, plugin có thể hiểu là là một phần mềm, hay nói cách khác nó là một hay nhiều tập tin PHP có chứa nhiều dòng lệnh hoặc hàm (function) do người dùng viết ra. Và dĩ nhiên nếu đã là các tập tin PHP thì plugin sẽ phải được viết với ngôn ngữ PHP, trong đó ngoài các hàm do người dùng tạo ra thì còn các hàm kế thừa từ thư viện có sẵn của WordPress.

Đối với WordPress, nếu cần một plugin nào đó thì chúng ta có thể tìm kiếm trước trên mạng thông qua Google hoặc tìm kiếm trực tiếp trên kho plugin của WordPress vì hiện tại kho plugin của WordPress hầu như đã đáp ứng đủ nhu cầu của người sử dụng.

Khi đến với kho plugin này, chúng ta sẽ dễ dàng tìm kiếm cho mình một plugin vừa ý bởi hệ thống có hỗ trợ đánh giá xếp hạng theo sự phổ biến của plugin hoặc sắp xếp plugin theo từng chức năng.

\* Tại sao lại phải dùng plugin

Nếu như xem WordPress là shop bán đồ chơi thì một trang blog WordPress mà không cài đặt plugin thì coi như shop đồ chơi đó chỉ có cái vỏ chứ không có đồ chơi hay bất cứ thứ gì được chứa đựng bên trong.

Có rất nhiều plugin với mục đích sử dụng khác nhau, nó cung cấp cho người dùng một số chức năng và tiện ích nhất định. Người dùng có thể tìm kiếm plugin để phục vụ một vài sở thích nào đó hoặc plugin đó sẽ làm cho việc quản lý blog tốt hơn.

\* Những kiến thức cần có trước khi bắt đầu.

+ Kiến thức về HTML và CSS cơ bản

+ Kiến thức về PHP cơ bản

+ Ngoài 2 loại kiến thức bên trên thì có thể bổ sung thêm kiến thức về MySQL, Javascript… nếu cần.

\* Công cụ cần thiết

+ Phần mềm hỗ trợ lập trình PHP (Đề nghị dùng **Notepad++** hoặc **Sublime Text** 2).

+ Đã cài đặt giả lập localhost (Đề nghị dùng **XAMP** hoặc **AppServ**) hoặc có **host**

+ Đã cài đặt sẵn một blog WordPress trên **localhost** hoặc **host**.

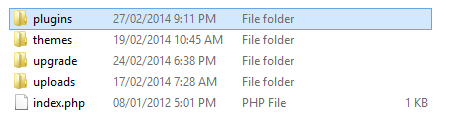
## Bắt đầu viết plugin

**\* Cấu trúc thư mục và tập tin**

Tạo thư mục và tập tin là hai bước cần phải làm đầu tiên trong quá trình viết plugin, nó giống như việc xây nhà vậy, chúng ta phải tạo ra được một cái móng cố định rồi sau đó mới tiếp tục xây tiếp phần bên trên, ở đây cấu trúc thư mục đóng vai trò tương tự vậy.

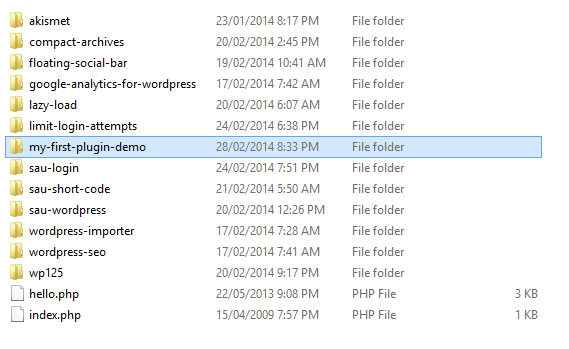
Thư mục của plugin phải được đặt trong thư mục mà WordPress dành riêng cho các plugin hoạt động, bắt buộc phải bỏ plugin vào thư mục này: **wp-content/plugins** thì mới có thể kích hoạt và sử dụng.

Như vậy, thư mục plugin phải được đặt bên trong thư mục plugins với đường dẫn như bên trên, sau đó các thư mục con và các tập tin sẽ được đặt nằm trong thư mục gốc plugin.



**\* Tên thư mục và tên của các tập tin**

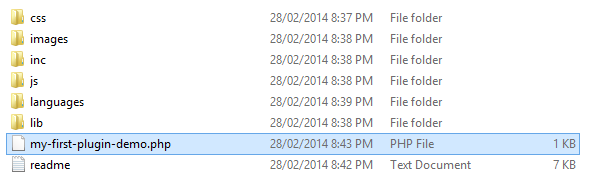
Tên của thư mục cũng là tên của plugin viết thường không có dấu Tiếng Việt và các từ được nối với nhau bằng dấu gạch ngang. Ví dụ như:**sssplugin-demo**. Tên của thư mục sẽ đại diện cho plugin, sau này nếu như cần truy vấn tới các tập tin hình ảnh, css hay javascript trong thư mục của plugin thì chúng ta sẽ phải sử dụng đến tên này, WordPress gọi tên của thư mục chứa plugin là **text-domain**.



Đối với tên của tập tin PHP cũng vậy, chúng không được đặt trùng tên với các plugin khác. Thông thường các lập trình viên sẽ lấy tên của thư mục chứa plugin đặt tên cho tập tin PHP chính của plugin. Một plugin phải có ít nhất một tập tin PHP, ngoài ra thì chúng ta có thể tạo thêm bao nhiêu tập tin PHP phụ cũng được.

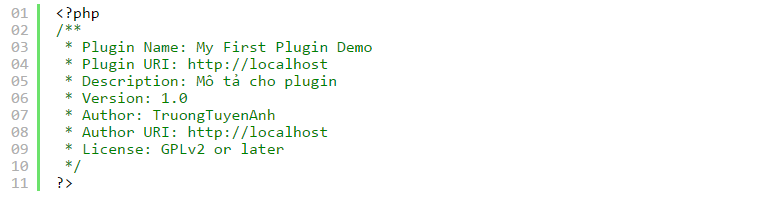
**\* Các tập tin cần thiết**

Đối với mỗi plugin thì chỉ cần có ít nhất 2 tập tin, đó là tập tin PHP chính của plugin và tập tin readme.txt chứa thông tin về tên, thông tin các phiên bản cũng như các thông tin liên quan khác đối với plugin. Nếu như chúng ta tự viết plugin với mục đích là tự dùng thì không cần tạo tập tin readme.txt.

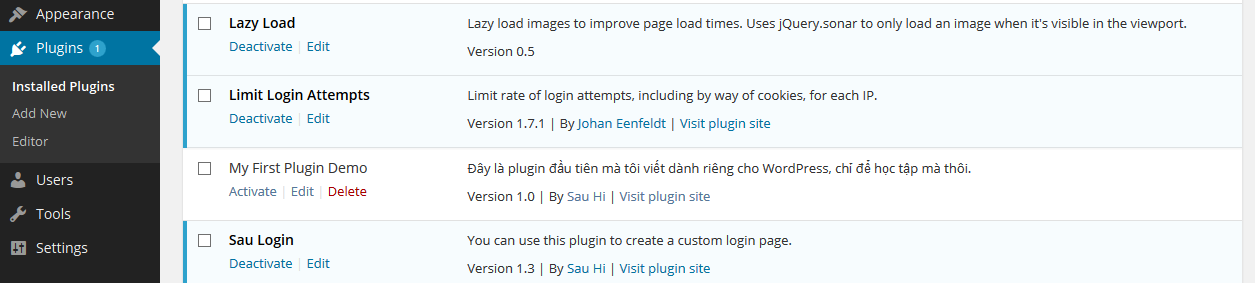


**Bắt đầu viết code cho plugin**

Đối với tập tin PHP chính của plugin, chúng ta phải đặt một vài ghi chú ở phía trên cùng của tập tin, những ghi chú này bao gồm **tên của plugin, tên của tác giả, mô tả cho plugin, phiên bản hiện tại của plugin và đường dẫn trang chủ của plugin với trang chủ của tác giả.** Cụ thể hơn, những phần cơ bản này sẽ phải tạo giống như bên dưới.



Vậy là bây giờ chúng ta có thể vào bảng điều khiển và kích hoạt plugin. Đây là một plugin hoàn toàn mới và nó mới chỉ có tên chứ chưa có nội dung bên trong.



## Tìm hiểu về Hook

**\* Hook là gì**

Đây là khái niệm do WordPress đưa ra để cho phép chúng ta thực thi một hàm nào đó vào một thời gian xác định. Có 2 loại **Hook** chính đó là: **Action** và **Filter**.

**\* Action là gì**

**Action** là một loại **Hook**, và nó là một hàm PHP. Action được đặt ở một ví trí nhất định cho phép chúng ta gọi một hàm khác tại vị trí đặt **Action**.

Ví dụ như chúng ta muốn thêm một đoạn mã javascript nào đấy vào phần footer trước khi kết thúc thẻ </body> thì có thể sử dụng **Action wp\_footer**.

**\* Filter là gì**

**Filter** cũng là một loại **Hook**, nó được đặt nằm giữa cơ sở dữ liệu và trình duyệt web, khi thực thi một câu lệnh nào đó thì đều phải thông qua ít nhất một Filter, và dĩ nhiên đây cũng là một hàm PHP. Filter đóng vai trò là người quản lý, nó sẽ lọc những thông tin người dùng muốn đưa vào cơ sở dữ liệu và ngược lại.

## Lập trình hướng đối tượng và cách đặt tên hàm

Trong một blog WordPress thì có thể sẽ được cài đặt rất nhiều plugin, trong mỗi plugin thì có rất nhiều hàm PHP. Để tránh trường hợp bị trùng lặp tên hàm thì chúng ta nên áp dụng lập trình hướng đối tượng vào đây bằng cách tạo ra các class riêng để sử dụng.

Đối với tên của class nên đặt theo tên của plugin,Ví dụ: **My\_First\_Plugin\_Demo**. Bên trong class này thì chúng ta có thể tạo mới nhiều hàm với chức năng khác nhau, do vậy sẽ dễ dàng kiểm soát được độ trùng lặp của mỗi hàm.

Nếu không sử dụng class thì chúng ta có thể áp dụng quy tắc đặt tên cho hàm bằng cách đặt tên viết tắt của plugin phía trước tên hàm. Ví dụ chúng ta có một hàm my\_function thì sẽ đặt tên là mfpd\_my\_function (với mfpd là chữ viết tắt của my first plugin demo) để tránh trùng lặp tên hàm với các plugin khác.

# XÂY DỰNG PLUGIN DEMO

## Phát biểu bài toán

- Trong khuôn khổ nghiên cứu cách viết plugin cho wordpress em chỉ dừng lại ở mức độ nghiên cứu cách viết plugin đơn giản đó là: viết plugin đơn giản về quản lý thông tin của một team work, các chức năng chỉ đơn giản là các thao tác thêm, sửa, xóa, hiển thị dữ liêu để nghiên cứu cách thức viết plugin cho wordpress, cách plugin hoạt động và cách làm việc với cơ sở dữ liệu trong wordpress.

- Thiết kế cơ sở dữ liệu cho bài toán:

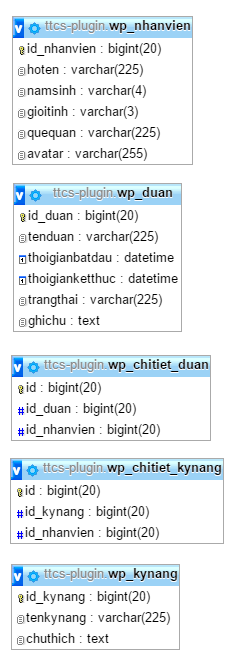
+ Bảng nhanvien: lưu thông tin về các thành viên của teamwork

+ Bảng duan: lưu thông tin về các dự án

+ Bảng kynang: lưu thông tin về các kỹ năng

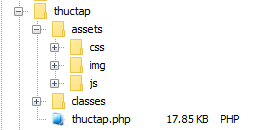
+ Bảng chitiet\_duan: lưu thông tin chi tiết về các dự án

+ Bảng chitiet\_kynang: lưu thông tin chi tiết về các kỹ năng



## Viết code cho plugin

1. Cấu trúc thư mục.



- thư mục: **thuctap** là thư mục chính lưu tất cả các code của plugin

- thư mục: **assets** là thư mục lưu các file về css, images, js

- thư mục: **classes** là thư mục lưu các file php cần thiết cho plugin

- file: **thuctap.php** là file chạy chính của plugin, file này cần trùng tên với thư mục cha: thuctap và khai báo thông tin của plugin: tên plugin, tác giả …

1. Khai báo thông tin chính cho plugin.

\* **Code cho file: thuctap.php**

- Khai báo thông tin về plugin

**<?php**

*/\**

*Plugin Name: Manage Teamwork*

*Description: Wordpress Plugin đơn giản quản lý thông tin một teamwork.*

*Plugin URI: https:localhost*

*Author URI: https:localhost*

*Author: Truong Tuyen Anh*

*License: Public Domain*

*Version: 1.0*

*Text Domain: simple\_plugin*

*\*/*

**?>**

**-** Trong đó:

+ Plugin Name: khai báo tên plugin

+ Description: Mô tả cho plugin: như chức năng, cách hoạt động …

+ Plugin URI: đường đẫn tới website của plugin

+ Author: Thông tin tác giả

+ Author URI: đường đẫn tới website của tác giả

+ License: Thông tin về giấy phép sử dụng plugin

+ Version: Phiên bản của plugin

+ Text Domain: khai báo chuỗi text sử dụng để dịch trong trường hợp hỗ trợ website đa ngôn ngữ

- Khai báo các hằng số cần dùng:

**<?php**

**define**( **TT\_DIR\_PATH**, **plugin\_dir\_path**( **\_\_FILE\_\_** ) ); //khai báo hằng TT\_DIR\_PATH lưu đường dẫn tuyệt đối tới thu muc của plugin này

**define**( **TT\_DIR\_URL**, **plugin\_dir\_url**( **\_\_FILE\_\_** ) ); //Khai báo hằng TT\_DIR\_URL lưu url của plugin này

**?>**

**-** Chèn các file cần thiết cho plugin

**<?php**

**require\_once**( **ABSPATH** . 'wp-admin/includes/upgrade.php' );//Cần chèn file này để có thể dùng hàm **dbDelta();**

**require\_once** **TT\_DIR\_PATH** . 'classes/class.TT\_KyNang.php'; //Class Ky Nang chứa các hàm xử lý thông tin liên quan tới kỹ năng

**require\_once** **TT\_DIR\_PATH** . 'classes/class.TT\_Nhanvien.php';//Class Nhan Vien chứa các hàm xử lý thông tin liên quan tới nhân viên

**require\_once** **TT\_DIR\_PATH** . 'classes/class.TT\_Duan.php';//Class Du An chứa các hàm xử lý thông tin liên quan tới dự án

if ( **!class\_exists**('WP\_List\_Table') ) {

**require\_once**(**ABSPATH** . 'wp-admin/includes/class-wp-list-table.php'); // chèn file này để có thể sử dụng lớp WP\_List\_Table được xây dựng sẵn trong WordPress Core

}

**?>**

- Chèn các file javascript, jquery, css cần thiết cho plugin

**<?php**

function **enqueue\_script**(){

**wp\_enqueue\_script**( 'jquery\_min', TT\_DIR\_URL . 'assets/js/jquery-1.12.3.min.js', array('jquery'), null, true );

**wp\_enqueue\_script**( 'jquery\_ui', TT\_DIR\_URL . 'assets/js/jquery-ui.min.js', array('jquery'), null, true );

**wp\_enqueue\_scrip**t( 'jquery\_choosen', TT\_DIR\_URL . 'assets/js/chosen.jquery.min.js', array('jquery'), null, true );

**wp\_enqueue\_script**( 'jquery\_function', TT\_DIR\_URL . 'assets/js/function.js', array('jquery'), null, true );

**wp\_enqueue\_script**( 'jquery-carousel', TT\_DIR\_URL . 'assets/js/owl.carousel.min.js', array( 'jquery' ), null, true );

**wp\_enqueue\_style**( 'jquery-ui-css', TT\_DIR\_URL . 'assets/css/jquery-ui.min.css', false, '' );

**wp\_enqueue\_style**( 'jquery-ui-theme-css', TT\_DIR\_URL . 'assets/css/jquery-ui.theme.min.css', false, '' );

**wp\_enqueue\_style**( 'jquery-ui-structure-css', TT\_DIR\_URL . 'assets/css/jquery-ui.structure.min.css', false, '' );

**wp\_enqueue\_style**( 'choosen.min.js', TT\_DIR\_URL . 'assets/css/chosen.min.css', false, '' );

**wp\_enqueue\_style**( 'fontawesome-css', TT\_DIR\_URL .'assets/css/font-awesome.min.css', false, '' );

**wp\_enqueue\_style**( 'carousel-css', TT\_DIR\_URL . 'assets/css/owl.carousel.css', false, '' );

**wp\_enqueue\_style**( 'custom-css', TT\_DIR\_URL . 'assets/css/custom.css', false, '' );

**wp\_enqueue\_media();**

}

**add\_action**( 'admin\_enqueue\_scripts', 'enqueue\_script' );

**?>**

Ở đây chúng ta sẽ sử dụng action hook **admin\_enqueue\_scripts** để chèn các file js, css vào trong khu vực trang admin. Trong plugin này em có sử dụng các thư viện: jquery, jquery-ui, jquery\_choosen, jquery owl-carousel, thư viện fontawesome.

-Cuối cùng là code cho lớp **TT\_Teamwork**:

+ Chúng ta sẽ tạo ra lớp **TT\_Teamwork:**

**<?php**

**class TT\_Teamwork**{

**public** $my\_db\_version = '1.0';//Biến lưu thông tin version hiện tại của database thiết kế cho plugin này

**function** \_\_construct(){

**register\_activation\_hook**( \_\_FILE\_\_, array( $this, 'create\_table' ) );//Đăng ký activation\_hook thông qua hàm create\_table để tạo ra các bảng dữ liệu cần thiết khi kích hoạt plugin

**register\_activation\_hook**( \_\_FILE\_\_, array( $this, 'dummy\_data' ) );//Đăng ký activation\_hook thông qua hàm dummy\_data để chèn dữ liệu mẫu vào các bảng plugin, tránh các lỗi không có dữ liệu

**register\_deactivation\_hook**( \_\_FILE\_\_, array( $this, 'delete\_table' ) );//Đăng ký deactivation\_hook để tiến hành xóa các bảng dữ liệu khi ngừng kích hoạt plugin

**add\_action**( 'admin\_menu', array( $this, 'register\_setting\_menu' ) );

**add\_action**( 'init', array( $this, 'tt\_load\_languages' ) );

}

**?>**

Trong đó: hàm **\_\_construct(),** là hàm khởi tạo, hàm này sẽ chạy đầu tiên khi lớp đối tượng được khởi tạo, do đó chúng ta sẽ sử dụng hàm này để đăng ký các **activation hook**, và **deactivation hook**, cũng như sử dụng các hook để chạy các hàm settings. Thông qua hàm **register\_activation\_hook**, 2 phương thức **create\_table, dummy\_data** sẽ được chạy để tạo ra các bảng dữ liệu dùng cho plugin và chèn vào dữ liệu mẫu để tránh các lỗi không mong muốn do chưa có dữ liệu. Phương thức **delete\_table** sẽ được chạy thông qua hook: **register\_deactivation\_hook** để tiến hành xóa các bảng dữ liệu khi plugin bị người dùng xóa từ bảng điều khiển plugin của wordpress.

+ Tiếp theo, vẫn trong lớp TT\_Teamwork chúng ta xây dựng thêm các phương thức sau:

\* Phương thức: **create\_table,** phương thức này đã được khai báo trong hook **register\_activation\_hook**:

**<?php**

**public function** create\_table(){

**global** $wpdb;

$query = "

CREATE TABLE {$wpdb->prefix}duan(

id\_duan BIGINT NOT NULL AUTO\_INCREMENT,

tenduan VARCHAR(225) NOT NULL,

thoigianbatdau DATETIME NOT NULL,

thoigianketthuc DATETIME NOT NULL,

trangthai VARCHAR(225) NOT NULL,

ghichu TEXT NULL,

PRIMARY KEY (id\_duan)

);

CREATE TABLE {$wpdb->prefix}nhanvien(

id\_nhanvien BIGINT NOT NULL AUTO\_INCREMENT ,

hoten VARCHAR(225) NOT NULL ,

namsinh VARCHAR(4) NOT NULL ,

gioitinh VARCHAR(3) NOT NULL ,

quequan VARCHAR(225) NOT NULL ,

avatar VARCHAR(255) NOT NULL,

PRIMARY KEY (id\_nhanvien)

);

CREATE TABLE {$wpdb->prefix}kynang(

id\_kynang BIGINT NOT NULL AUTO\_INCREMENT,

tenkynang VARCHAR(225) NOT NULL,

chuthich TEXT NULL,

PRIMARY KEY (id\_kynang)

);

CREATE TABLE {$wpdb->prefix}chitiet\_duan(

id BIGINT NOT NULL AUTO\_INCREMENT,

id\_duan BIGINT NOT NULL,

id\_nhanvien BIGINT NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE {$wpdb->prefix}chitiet\_kynang(

id BIGINT NOT NULL AUTO\_INCREMENT,

id\_kynang BIGINT NOT NULL,

id\_nhanvien BIGINT NOT NULL,

PRIMARY KEY (id)

);

";

**dbDelta**( $query );

}

**?>**

Trong đó chúng ta sử dụng biến **global** $wpdb có sẵn của wordpress để tương tác với cơ sở dữ liệu. Ở đây **$query** chúng ta sẽ gán query SQL tạo ra 5 bảng dữ liệu phục vụ cho plugin. Cuối cùng là gọi hàm **dbDelta()** để thực thi query SQL tạo bảng dữ liệu.

\* Phương thức: **dummy\_data()**, thêm dữ liệu mẫu vào các bảng vừa tạo

**<?php**

**public function** dummy\_data(){

**global** $wpdb;

//Dữ liệu mẫu cho bảng duan

**$wpdb->insert**( $wpdb->prefix . 'duan', array(

'id\_duan' => 1,

'tenduan' => 'Website bán hàng cho công ty Incomtech',

'thoigianbatdau' => '2015-10-16',

'thoigianketthuc' => '2015-11-16',

'trangthai' => 'Đã hoàn thành',

'ghichu' => 'Website bán hàng nội thất nhựa'

));

//du lieu mau cho bang \_nhanvien

**$wpdb->insert**( $wpdb->prefix . 'nhanvien', array(

'id\_nhanvien' => 1,

'hoten' => 'Nguyễn Văn An',

'namsinh' => '1990',

'gioitinh' => "Nam",

'quequan' => 'Thái Bình',

));

//du lieu mau cho bang \_kynang

**$wpdb->insert**( $wpdb->prefix . 'kynang', array(

'id\_kynang' => 1,

'tenkynang' => 'HTML',

'chuthich' => 'Thành thạo HTML',

));

//du lieu mau cho bang: \_chitiet\_duan

**$wpdb->insert**( $wpdb->prefix . 'chitiet\_duan', array(

'id' => 1,

'id\_duan' => 1,

'id\_nhanvien' => 1,

));

//du lieu cho bang \_chitiet\_kynang

**$wpdb->insert**( $wpdb->prefix . 'chitiet\_kynang', array(

'id' => 1,

'id\_kynang' => 1,

'id\_nhanvien' => 1,

));

}

**?>**

Phương thức này có sử dụng phương thức **insert()** của biến **global $wpdb** để tiến hành insert dữ liệu vào các bảng.

\* Phương thức: **delete\_table**

**<?php**

**public function** delete\_table(){

**global** $wpdb;

$sql = "DROP TABLE IF EXISTS `{$wpdb->prefix}duan`, `{$wpdb->prefix}nhanvien`, `{$wpdb->prefix}kynang`, `{$wpdb->prefix}chitiet\_duan`, `{$wpdb->prefix}chitiet\_kynang`";

$wpdb->query( $sql );

$this->del\_db\_version();

}

**?>**

Phương thức này chỉ đơn giản làm nhiệm vụ xóa hết các bảng đã tạo ra trong phương thức **create\_table**()

\* Phương thức: **register\_setting\_menu(),** phương thức này được gọi qua hook **admin\_menu** để tạo ra menu setting cho plugin này:

**<?php**

**public function** register\_setting\_menu(){

**add\_menu\_page**( \_\_("TT Teamwork", "simple\_plugin"), \_\_("TT Teamwork","simple\_plugin"), "activate\_plugins", "tt\_teamwork", array( $this, "tt\_teamwork\_callback" ) );

**add\_submenu\_page**( "tt\_teamwork", \_\_( "Danh sách nhân viên","simple\_plugin" ), \_\_( "Danh sách nhân viên","simple\_plugin" ), "activate\_plugins", "ds\_nhanvien", array( "TT\_Nhanvien", "tt\_page\_nhanvien\_callback" ) );

**add\_submenu\_page**( "tt\_teamwork", \_\_( "Thêm mới nhân viên", "simple\_plugin" ), \_\_( "Thêm mới nhân viên", "simple\_plugin" ), "activate\_plugins", "new\_nhanvien", array( "TT\_Nhanvien", "tt\_new\_nhanvien\_callback" ) );

**add\_submenu\_page**( 'tt\_teamwork', \_\_( "Danh sách kỹ năng", "simple\_plugin" ), \_\_( "Danh sách kỹ năng", "simple\_plugin" ), "activate\_plugins", "ds\_ky\_nang", array( "TT\_KyNang", "tt\_kynang\_page\_callback" ) );

**add\_submenu\_page**( 'tt\_teamwork', \_\_( "Thêm mới kỹ năng", "simple\_plugin" ), \_\_( "Thêm mới kỹ năng", "simple\_plugin" ), "activate\_plugins", "new\_kynang", array( "TT\_KyNang", "tt\_new\_kynang\_callback" ) );

**add\_submenu\_page**( 'tt\_teamwork', \_\_( "Danh sách dự án", "simple\_plugin" ), \_\_( "Danh sách dự án", "simple\_plugin" ), "activate\_plugins", "ds\_duan", array( "TT\_Duan", "tt\_duan\_page\_callback") );

**add\_submenu\_page**( 'tt\_teamwork', \_\_( "Thêm mới dự án", "simple\_plugin" ), \_\_( "Thêm mới dự án", "simple\_plugin" ), "activate\_plugins", "new\_duan", array( "TT\_Duan", "tt\_new\_duan\_page\_callback") );

}

**?>**

Ở đây chúng ta sẽ sử dụng hàm: **add\_menu\_page()** để đăng ký một menu mới trong khu vực admin, hàm này có 7 tham số **add\_menu\_page**( $page\_title, $menu\_title, $capability, $menu\_slug, $function, $icon\_url, $position );

Trong đó:

+ $page\_title: tiêu đề của page,

+ $menu\_title: tiêu đề của menu,

+ $capability là quyền của user, trong trường hợp này những user có quyền được khai báo mới có thể nhìn thấy menu này,

+ $menu\_slug: chuỗi đường dẫn cho menu

+ $function: hàm sẽ xử lý menu này

+ $icon\_url: sử dụng icon tùy chọn cho menu này, mặc định sẽ chọn icon mặc định của wordpress

+ $position: vị trí hiển thị menu.

+ Các thông tin chi tiết về hàm và các tham số được đặt tại: https https://developer.wordpress.org/reference/functions/add\_menu\_page/

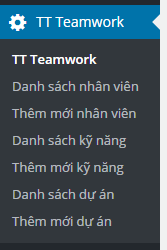
Hàm **add\_submenu\_page** ( $parent\_slug,  $page\_title,  $menu\_title,  $capability,  $menu\_slug, $function ); được sử dụng để thêm một submenu vào meu cha

+$parent\_slug: là chuỗi slug của menu cha cần thêm sub menu

+ các tham số còn lại tương tự như hàm: **add\_menu\_page**

**+** Thông tin chi tiết về hàm này đặt tại: https://developer.wordpress.org/reference/functions/add\_submenu\_page/

Kết quả khi chạy hàm này chúng ta sẽ có một menu như sau:



\* Xây dựng phương thức **tt\_teamwork\_callback(),**  đây là phương thức mà chúng ta khai báo trong hàm **add\_menu\_page()**, hàm này sẽ có nhiệm vụ xử lý phần hiển thị của menu chính mà chúng ta đã đăng ký.

**<?php**

**public function** tt\_teamwork\_callback(){ ?>

<div class="wrap">

<div class="team\_member" >

<?php $num\_member = self::tt\_count\_total\_members(); ?>

<h2**><?php** \_e( 'Tất cả các thành viên: ('. $num\_member .')', 'simple\_plugin' ); **?>**</h2>

**<?php** self::tt\_get\_team\_member(); **?>**

</div>

<div class="team\_projects">

**<?php** $num\_project = self::tt\_count\_total\_projects(); **?>**

<h2**><?php** \_e( "Tất cả các dự án: ({$num\_project})", "simple\_plugin" ); **?>**</h2>

**<?php** self::tt\_get\_project\_status(); **?>**

</div>

</div>

**<?php**  } **?>**

Và xây dựng các phương thức được gọi trong phương thức này:

**<?php**

**public static function** tt\_selected( $select, $value ){

if( isset( $select ) && !empty( $select ) ){

if( $select == $value){

echo 'selected="selected"';

}

}

}

**public static function** tt\_checked( $check, $value ){

if( isset( $check ) && !empty( $check ) ){

if( $check == $value){

echo 'checked="checked"';

}

}

}

**public static function** tt\_count\_total\_members(){

global $wpdb;

$table\_name = $wpdb->prefix . 'nhanvien';

$all\_member = $wpdb->get\_var( "SELECT COUNT(\*) FROM {$table\_name}" );

if( is\_numeric( $all\_member ) ){

return $all\_member;

}else{

return 0;

}

}

**public static function** tt\_count\_total\_projects(){

global $wpdb;

$table\_name = $wpdb->prefix . 'duan';

$all\_projects = $wpdb->get\_var( "SELECT COUNT(\*) FROM {$table\_name}" );

if( is\_numeric( $all\_projects ) ){

return $all\_projects;

}else{

return 0;

}

}

**public static function** tt\_get\_team\_member(){

global $wpdb;

$table\_name = $wpdb->prefix . 'nhanvien';

$all\_member = $wpdb->get\_results( "SELECT \* FROM $table\_name", ARRAY\_A );

if( **is\_array**( $all\_member ) && !**empty**( $all\_member ) ){

echo '<div class="team\_member owl-carousel" data-nav="true" data-autoplay="false" data-dots="false" data-loop="true" data-margin="10" data-responsive=\'{"0":{"items":4},"600":{"items":4},"1000":{"items":4}}\' >';

**foreach**( $all\_member as $key=>$value ){

$nhanvien\_kynang = TT\_Nhanvien::tt\_get\_selected\_detail\_kynang( $value['id\_nhanvien'] );

$nhanvien\_duan = TT\_Nhanvien::tt\_get\_selected\_detail\_duan( $value['id\_nhanvien'] );

**?>**

<div class="each\_member">

<div class="member\_avatar">

**<?php** if( **!empty**( $value['avatar'] ) ){

echo '<img src="'. $value['avatar'] .'" />';

}else{

self::tt\_default\_avatar( $value['gioitinh'] );

} **?>**

</div>

<div class="member\_info">

**<?php** **if**( $value['hoten'] ): **?>**<p class="member-name"><strong>Họ Tên:</strong> **<?php** echo '<a href="?page=new\_nhanvien&id\_nhanvien=' . $value['id\_nhanvien'] . '">' . esc\_html( $value['hoten'] ) . '</a>'; **?>**</p**><?php endif; ?>**

**<?php if**( $value['namsinh'] ): **?>**<p class="member-dateofbirth"><strong>Năm sinh:</strong> **<?php** **echo** esc\_html( $value['namsinh'] ); **?>**</p**><?php** endif; **?>**

**<?php if**( $nhanvien\_kynang ): **?>**<p class="member-skills"><strong>Các kỹ năng:</strong> **<?php** echo $nhanvien\_kynang; **?>**</p**><?php endif; ?>**

**<?php if**( $nhanvien\_duan ): **?>**<p class="member-projects"><strong>Các dự án:</strong> **<?php echo** $nhanvien\_duan; **?>**</p**><?php endif; ?>**

</div>

<div class="clearfix"></div>

</div>

**<?php**

}

**echo** "</div>";

}

}

**public static function** tt\_get\_member\_joined\_project( $project\_id ){

**global** $wpdb;

$table\_name = $wpdb->prefix . 'chitiet\_duan';

$list\_member = $wpdb->get\_results( $wpdb->prepare( "SELECT id\_nhanvien FROM $table\_name WHERE id\_duan = %d", $project\_id ), ARRAY\_A );

**if( is\_array**( $list\_member ) && **!empty**( $list\_member ) **)**{

echo "<ol>";

**foreach**( $list\_member as $key=>$value ){

**echo** "<li>";

self::tt\_get\_nhanvien\_name( $value['id\_nhanvien'] );

**echo** "</li>";

}

**echo** "</ol>";

}**else**{

**echo** "Không có thành viên nào tham gia dự án này!";

}

}

**public static function** tt\_get\_nhanvien\_name( $id\_nhanvien ){

**global** $wpdb;

$table\_name = $wpdb->prefix . "nhanvien";

$nhanvien = $wpdb->get\_results( $wpdb->prepare( "SELECT id\_nhanvien,hoten FROM {$table\_name} WHERE id\_nhanvien = %d", $id\_nhanvien ), ARRAY\_A );

**if( is\_array**( $nhanvien ) && !empty( $nhanvien ) **)**{

**echo** sprintf( '<a href="?page=new\_nhanvien&id\_nhanvien=%d">%s</a>', $nhanvien[0]['id\_nhanvien'], $nhanvien[0]['hoten'] );

}**else**{

**echo** \_\_( "Không có dữ liệu", "simple\_plugin" );

}

}

**public static function** tt\_get\_project\_status(){

**global** $wpdb;

$table\_name = $wpdb->prefix . 'duan';

$all\_member = $wpdb->get\_results( "SELECT \* FROM $table\_name", ARRAY\_A );

**if( is\_array**( $all\_member ) && !empty( $all\_member ) **)**{

**echo** '<div class="team\_project owl-carousel" data-nav="true" data-autoplay="false" data-dots="false" data-loop="true" data-margin="10" data-responsive=\'{"0":{"items":4},"600":{"items":4},"1000":{"items":4}}\' >';

**foreach**( $all\_member as $key=>$value ){

**?>**

<div class="each\_member">

<div class="prject\_preview">

<img src=" **<?php echo** esc\_url( TT\_DIR\_URL . '/assets/img/project\_thumb.jpg' ); **?>**" />

</div>

<div class="project\_info">

**<?php if**( $value['tenduan'] ): **?>**<p class="project-name"><strong>Tên dự án:</strong> **<?php echo** '<a href="?page=new\_duan&id\_duan=' . $value['id\_duan'] . '">' . esc\_html( $value['tenduan'] ) . '</a>'; **?>**</p**><?php endif; ?>**

**<?php if**( $value['thoigianbatdau'] ): **?>**<p class="start\_time"><strong>Thời gian:</strong> **<?php echo** esc\_html( date( 'd-m-Y', strtotime( $value['thoigianbatdau'] ) ) ); **?>** <i class="fa fa-long-arrow-right"></i> **<?php if**( !empty( $value['thoigianketthuc'] ) ){ **echo** **date**( 'd-m-Y', **strtotime**( $value['thoigianketthuc'] ) ); } **?>**</p**><?php endif; ?>**

**<?php if**( $value['trangthai'] ): **?>**<p class="projects-status"><strong>Trạng thái:</strong> **<?php echo** $value['trangthai']; **?>**</p**><?php endif; ?>**

<p class="joined\_member">

<strong>Các thành viên tham gia:</strong>

**<?php** $list\_ids = self::tt\_get\_member\_joined\_project( $value['id\_duan'] ); **?>**

</p>

</div>

<div class="clearfix"></div>

</div>

<?php

}

**echo** "</div>";

}

}

**public static function** tt\_default\_avatar( $gender = "Nam" ){

if( $gender == "Nữ" || $gender == "nữ" || $gender == "nu" || $gender == "Nu" ){

echo '<img src="' . TT\_DIR\_URL . 'assets/img/female-avatar.jpg'. '" />';

}else{

echo '<img src="' . TT\_DIR\_URL . 'assets/img/male-avatar.jpg'. '" />';

}

}

\* Cuối cùng chúng ta thêm phương thức **tt\_load\_languages**: để load file ngôn ngữ phục vụ cho việc dịch chuỗi trong hỗ trợ web đa ngôn ngữ.

**<?php**

**public function** tt\_load\_languages(){

load\_plugin\_textdomain( 'simple\_plugin', false, **dirname**(plugin\_basename(\_\_FILE\_\_) ) . ' /languages'); //hàm này load file ngôn ngữ trong thư mục ngôn ngữ

}

**?>**

Để chạy class TT\_Teamwork chúng ta cần khởi tạo nó

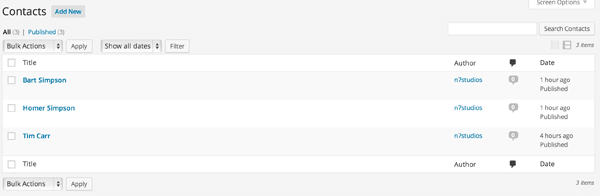
**<?php** new TT\_Teamwork(); **?>**

Lúc này lớp TT\_Teamwork sẽ chạy các hàm được chỉ định cho ra kết quả như mong muốn

1. Xây dựng các lớp thực hiện thao tác: thêm, sửa, xóa …
2. Tìm hiểu về class WP\_List\_Table của wordpress

\* Giới thiệu:

- Trong wordpres class WP\_List\_Table được sử dụng để hiển thị dữ liệu dưới dạng bảng. Ví dụ như bảng: tất cả các post, tất cả các page … trong admin page.



\* Các phương thức cần quan tâm:

+ Phương thức: **column\_default()**

**+** Phương thức: **column\_{ tên cột }():** khai báo cột sẽ được hiển thị

**+** Phương thức: **column\_cb():** tạo ra check box cho các dòng dữ liệus

+ Phương thức: **get\_columns():**lấy danh sách các cột sẽ được hiển thị ra bảng

+ Phương thức: **get\_sortable\_columns():** lấy ra các cột sẽ được sử dụng cho sắp xếp

+ Phương thức: **get\_bulk\_actions():** lấy ra các thao tác sẽ thực hiện với nhiều bản ghi: ví dụ xóa các bản ghi

+ Phương thức: **process\_bulk\_action()**: thực hiện các thao tác đã khai báo ở phương thức: **get\_bulk\_actions()**

+ Phương thức: **prepare\_items():** chuẩn bị danh sách các phần từ sẽ được hiển thị

+ Thông tin chi tiết về các phương thức của lớp này được lưu tại: https://core.trac.wordpress.org/browser/tags/4.5/src//wp-admin/includes/class-wp-list-table.php. phiên bản wordpress tính tới thời điểm bản báo cáo này là 4.5

1. File classes/class.TT\_KyNang.php

\* Chúng ta khai báo lớp TT\_KyNang kế thừa lớp WP\_List\_Table để có thể sử dụng các phương thức được xây dựng sẵn của lớp WP\_List\_Table

**<?php**

***class* TT\_KyNang *extends* WP\_List\_Table**{

**function** \_\_construct(){

//Hàm khởi tạo của lớp

}

**public function** column\_default( $item, $colum\_name ){

}

**public function** colum\_id\_kynang( $item ){

//Khai báo cột **id\_kynang**

}

**public function** colum\_chuthich( $item ){

//Khai báo cột **chuthich**

}

**public function** column\_tenkynang( $item ){

//Khai báo cột **tenkynang**

}

**public function** column\_cb( $item ){

//tạo ra các ô checkbox cho mỗi dòng dữ liệu

}

**public function** get\_columns(){

//lấy ra các cột sẽ hiển thị

}

**public function** get\_sortable\_columns(){

//Lấy ra các cột dữ liệu được sắp xếp

}

**public function** get\_bulk\_actions(){

//Khai báo các thao tác sẽ thực hiện trên nhiều bản ghi

}

**public function** process\_bulk\_action(){

//phương thức xử lý các thao tác khai báo ở get\_bulk\_actions()

}

**public function** prepare\_items(){

//Lấy dữ liệu để hiển thị

}

**public function** tt\_kynang\_page\_callback(){

//phương thức xử lý trang danh sách tất cả các kỹ năng

}//end tt\_kynang\_page\_callback()

**public function** tt\_new\_kynang\_callback(){

//Phương thức xử lý thêm mới một kỹ năng hoặc update kỹ năng đã có

}

**public static function** tt\_kynang\_validate\_data( $item ){

//Hàm này dùng để validate dữ liệu trước khi nó được thêm vào database

}

}

**?>**

1. File classes/class.TT\_Duan.php

\* Tương tự như lớp TT\_KyNang thì lớp TT\_Duan chúng ta cũng sẽ kế thừa từ lớp WP\_List\_Table

**<?php**

**class TT\_Duan extends WP\_List\_Table{**

**public function** \_\_construct(){

//Hàm khởi tạo

}

**function** column\_default( $item, $column\_name ){

}

**function** column\_id\_duan( $item ){

//Khai báo cột id\_duan

}

**function** column\_trangthai( $item ){

//Khai báo cột trangthai

}

**function** column\_thoigianbatdau( $item ){

//Khai báo cột thoigianbatdau

}

**function** column\_thoigianketthuc( $item ){

//Khai báo cột thoigianketthuc

}

**function** column\_ghichu( $item ){

//Khai báo cột ghichu

}

**function** column\_tenduan( $item ){

//Khai báo cột tenduan

}

**function** column\_cb( $item ){

//Tạo các ô checkbox cho mỗi dòng dữ liệu

}

**function** get\_columns(){

//lấy các cột sẽ được hiển thị

}

**function** get\_sortable\_columns(){

//lấy ra các cột phục vụ cho việc sắp xếp dữ liệu khi hiển thị

}

**function** get\_bulk\_actions() {

//khai báo các thao tác sẽ thực hiện trên nhiều bản ghi

}

**function** process\_bulk\_action(){

//Xử lý các bulk\_actions được khai báo ở phương thức get\_bulk\_actions

}

**function** prepare\_items(){

//Lấy dữ liệu để hiển thị

}

**public function** tt\_duan\_page\_callback(){

//Thực hiện hiển thị bảng các dự án

}

**public function** tt\_new\_duan\_page\_callback(){

//Thực hiện chức năng thêm mới một dự án

}

**public static function** tt\_validate\_data\_duan( $item ){

//kiểm tra và lọc dữ liệu trước khi thêm vào database

}

}

**?>**

1. File classes/class.TT\_Nhanvien.php

\* Cũng tương tự như lớp TT\_KyNang, TT\_Duan thì lớp TT\_Nhanvien chúng ta cũng sẽ kế thừa từ lớp WP\_List\_Table

**<?php**

***class* TT\_Nhanvien *extends* WP\_List\_Table**{

**public function** \_\_construct(){

//Hàm khởi tạo

}

**function** column\_default( $item, $column\_name ){

}

**function** column\_id\_nhanvien( $item ){

//khai báo cột id\_nhanvien

}

**function** column\_namsinh( $item ){

//Khai báo cột namsinh

}

**function** column\_gioitinh( $item ){

//Khai báo cột gioitinh

}

**function** column\_cac\_duan( $item ){

//Khai báo cột cac\_duan

}

**function** column\_cac\_kynang( $item ){

//Khai báo cột cac\_kynang

}

**function** column\_quequan( $item ){

//Khai báo cột quequan

}

**function** column\_hoten( $item ){

//Khai báo cột hoten

}

**function** column\_cb( $item ){

//Tạo checkbox cho mỗi dòng dữ liệu

}

**function** get\_columns(){

//Lấy các cột sẽ được hiển thị

}

**function** get\_sortable\_columns(){

//khai báo các cột dùng để sắp xếp dữ liệu

}

**function** get\_bulk\_actions() {

//khai báo các hành động thực hiện với nhiều bản ghi

}

**function** process\_bulk\_action(){

//Xử lý các hành động được khai báo ở get\_bulk\_actions()

}

**function** prepare\_items(){

//Lấy dữ liệu để hiển thị ra ngoài

}

**public static function** tt\_get\_selected\_detail\_duan( $id\_nhanvien ){

//Lấy thông tin các dự án mà nhân viên này đã, đang tham gia thông qua id\_nhanvien

}

**public static function** tt\_get\_selected\_detail\_kynang( $id\_nhanvien ){

//Lấy thông tin chi tiết các kỹ năng của nhân viên thông qua id\_nhanvien

}

**public function** tt\_page\_nhanvien\_callback(){

//Hiển thị bảng dữ liệu

}

**public function** tt\_new\_nhanvien\_callback(){

//Xử lý việc thêm mới một nhân viên

}

**public static function** tt\_validate\_data\_nhanvien( $item ){

//validate các đầu vào

}

}

**?>**

\* Do lượng code của từng phương thức trong các lớp này rất nhiều nên em chỉ trình bày tên các phương thức sẽ xây dựng, còn nội dung từng file em đính kèm trong source code đi kèm bản báo cáo này, hoặc tại địa chỉ: https://github.com/TruongTuyen/Official-Plugin